# USING THE LTSS-DDA PROVIDER UPLOAD API

Version History

| Version | Date | Changes |
|---|---|---|
| V1.0 | 1/4/2019 | Original copy |
| V1.1 | 10/25/2019 | Updated for Testing Site |
| V2.0 | 01/08/2020 | Updated URL's and instructions |
| V2.1 | 5/26/2020 | Updated instrructions |
| V3.0 | 7/23/2020 | Updated to SSO v4 and updated URL's |
| V 4.0 | 7/23/2020 | Updated error code 1200 |

# Overview

The provider upload API is the primary way for the provider system to create billing entries in the Provider Portal platform. It's an HTTP based API that different provider apps and systems can use programmatically to POST billing entries.

# Using the Provider Upload API

**HTTP/1.1**

All data transfer conform to HTTP/1.1, and all endpoints require HTTPS.

**Test URLs**

- URL for authentication: https://provtestsso.ltssmaryland.org/connect/token
- URL for sending data: https://provtest.ltssmaryland.org/ltssv2/Ltss.SelfServeApi.Web/api/providerupload/uploadbillingentry

**Production URLs**

- URL for authentication: https://sso.ltssmaryland.org/connect/token
- URL for sending data: https://ltssmaryland.org/ltssv2/Ltss.SelfServeApi.Web/api/providerupload/uploadbillingentry

# Authentication

Authentication allows provider systems to create a billing entry using the Provider Upload API. Authentication also allows us to identify the provider system and the type of data being transmitted. Provider Upload API requires a JWT (JSON Web Token) each time you access the upload billing entry endpoint.

Authentication is the first request a system should make before beginning their upload process.

Please request MDH to provide the following provider specific credentials before using the Provider Upload API:

- AppId
- AppSecret
- SSO Username

***Note:*** *SSO accounts for accessing Provider Upload API have different roles and provider cannot use the same account for accessing Provider Portal and vice-versa.*

## Authentication Request

| | Key | Value | Comments |
|---|---|---|---|
| **Headers** | | | |
| | Authorization | Basic {Base64Encoding({appId}:{appSecret})<br><br>**See Note Below | appId and appSecret will be issued to each provider by MDH. The header value must be base64 encoded |
| | cache-control | no-cache | |
| | content-type | application/x-www-form-urlencoded | |
| | | | |
| **Post Body : x-www-form-urlencoded** | | | |
| | grant_type | password | Static value |
| | Username | {sso_username} | SSO Username will be provided by MDH |
| | password | {sso_password} | Providers will have the option reset their password. |

****Basic {Base64Encoding({appId}:{appSecret})**

Basic e2FwcElkfTp7YXBwU2VjcmV0fQ==

How to base64 encode the username and password:

**Windows:**

[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes('{appId}:{appSecret}'))

Where the output of the command is

e2FwcElkfTp7YXBwU2VjcmV0fQ==

**Linux:**

echo -n '{appId}:{appSecret}' | base64 -w9999

Where the output of the command is

e2FwcElkfTp7YXBwU2VjcmV0fQ==

## Sample Authentication Request in cURL

```
curl -X POST \
  https://provtestsso.ltssmaryland.org/connect/token \
  -H 'Authorization: Basic e2FwcElkfTp7YXBwU2VjcmV0fQ==' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'cache-control: no-cache' \
  -d 'grant_type=password&username={sso_username}&password={sso_password}'
```

Alternately, using the Forms options of curl:

```
curl -X POST \
  https://provtestsso.ltssmaryland.org/connect/token \
  -H 'Authorization: Basic e2FwcElkfTp7YXBwU2VjcmV0fQ==' \
  -H 'cache-control: no-cache' \
  -F grant_type=password \
  -F username={sso_username} \
  -F password={sso_password}
```

*Note: The above sample request is for illustration purpose only. Every programming language will need to build the request in its native implementation.*

### Authentication Response

| **Success** | | |
|---|---|---|
| | HTTP Response Code | 200 |
| Response Body - Sample | | |
| {<br>"access_token": "{TOKEN_STRING}",<br>"expires_in": {TOKEN_EXPIRATION_IN_SECONDS},<br>"token_type": "Bearer",<br>“scope”: “{TOKEN_SCOPE}”<br>} | | |
| **Fail** | | |
| Unauthorized | HTTP Response Code | 401 |
| {"error":"invalid_scope"} | Scopes Configured for Client do not allow token issuance | |
| {"error":"invalid_client"} | Authorization contents are invalid | |

### Authentication Response Body Attributes

| **Response Body** | | |
|---|---|---|
| | access_token | **JWT token string. This token string needs to be used in every request for the upload process.** |
| | expires_in | Token expiration time in seconds. Default value is 3600 seconds |
| | token_type | Specifies which token type is being returned. Token_type does not need to be used in subsequent calls.<br><br>Token type would always be **Bearer** |
| | scope | The scope in which this token is valid |

# Upload Billing Entry

After successful authentication, the provider  can access the Provider Upload Web API resource to upload billing entries one at a time. The Web API is designed to keep the small pay load for better throughput.

### Idempotent Design

Uploading a billing entry is idempotent by design. Each billing entry requires a unique transaction ID. If for some reason the client experiences an exception such as timeout or connection closed, the client system should use the same transaction ID for the retry call. This will avoid a duplicate billing entry in the Provider Portal system. You may reuse the same transaction ID when a request fails due to validation error.

### Upload Billing Entry Endpoint

https://provtest.ltssmaryland.org/ltssv2/Ltss.SelfServeApi.Web/api/providerupload/uploadbillingentry

### Upload Billing Entry Request Data Format

| Field | JSON Type | Required | Description |
|---|---|---|---|
| TransactionId | String | **Yes** | A unique transactionId for every request. A 128 bit UUID/GUID to avoid collision. |
| ServiceIdentifier | string | **Yes** | Alpha-numeric<br>Abbreviated service names/codes titles will be provided by DDA |
| ServiceDate | string | **Yes** | Valid Date of Service<br>Not greater than current date<br>Not older than 365 days<br>Should be greater than or equal to client's pilot date OR provider's billing phase-in date<br>mm-dd-yyyy format |
| ClientLtssId | string | **Yes** | Alpha-numeric<br>Valid Client's LTSS ID |
| ProviderMa | string | **Yes** | Alpha-numeric<br>Valid Provider MA Number |
| Units | number | **Yes** | Required if Cost is not specified<br>Number of Units<br>A whole number<br>No decimal is allowed |
| Cost | number | **Yes** | Required if Unit is not specified<br>Cost of Activity<br>up-to 2 decimal points |

## Data Contract for Upload Request

| | Key | Value | Comments |
|---|---|---|---|
| **Headers** | | | |
| | Authorization | Bearer {TOKEN_STRING} | Use **access_token** string from the authentication response. This header is required for every upload request. |
| | content-type | application/json | |
| | | | |
| **Post Body : application/json - Sample** | | | |

```
{
        "TransactionId": "00000000-0000-0000-0000-000000000000"
        "ServiceIdentifier": "xxxxxxxxxxxxxxx",
        "ServiceDate": "9/12/2018",
        "ClientLtssId": "A1234567890",
        "ProviderMa": "A12345678",
        "Units": 2,
        "Cost": 5.4
}
```

### Sample Billing Entry Request in cURL

```
curl -X POST \
  https://provtest.ltssmaryland.org/ltssv2/Ltss.SelfServeApi.Web/api/providerupload/uploadbillingentry
\
  -H 'Authorization': 'Bearer {{ selfserve_api_token }}' \
  -H 'Content-Type': 'application/x-www-form-urlencoded' \
  -H 'cache-control': 'no-cache' \
  -d '{ "TransactionId": "00000000-0000-0000-0000-000000000000", "ServiceIdentifier":
"xxxxxxxxxxxxxxx", "ServiceDate": "9/12/2018", "ClientLtssId": "A1234567890", "ProviderMa":
"A12345678", "Units": 2, "Cost": 5.4 }'
```

*Note: The above sample request is for illustration purpose only. Every programming language will need to build the request in its native implementation.*

**Upload Billing Entry Response**

| **Success** | | | |
|---|---|---|---|
| | HTTP Response Code | 200 | Success |
| Response Header | TID | {TransactionId} | The same transaction Id from Request. |
| **Fail** | | | |
| Response Header | TID | {TransactionId} | The same transaction Id from Request. |
| | HTTP Response Code | 401 | Unauthorized<br>This can be due to invalid or expired JWT. |
| | | 400 | Bad request<br>Data validation failed. 400 response will have ErrorCode and descriptive message for the validation that failed. |

**Response Body for HTTP Response code 400 : application/json – Sample**

```
{
        "Message": ""
        "ErrorCode":
}
```

**Error Codes**

## HTTP Status 400

The execution sequence of validations are in the order of Error codes. However, the Provider Upload API will short circuit the validations and return an HTTP 400 error upon first failed validation.

API at any point would always return a single error code and message.

| **Explanation of all Error codes and messages for HTTP response code 400** | | |
|---|---|---|
| **Error Code** | **Error Message** | **Validation Rule** |
| 1000 | Invalid SSO Username | No staff profile is associated with SSO username in LTSS |
| 1100 | TransactionId not present | Transaction ID is missing in request |
| 1200 | Invalid Date of Service | Date is not present in request<br>Date entered is invalid and cannot be parsed to a standard date (i.e. date is not in correct format)<br>Date is less than dda-go-live (configured) date<br>Difference between Service date and current date is more than 365 days<br>Date is less than the provider's billing phase-in date [OR] less than client's pilot date |

| | | |
|---|---|---|
| 1300 | Invalid Unit or Cost | Cost and Units are absent in request<br>Or<br>Both Cost and Units are specified and greater than 0 in request<br>Provided cost for unit-based service, or units for cost based service<br>Unit should be a whole number<br>Cost supports only up to 2 decimal point |
| 1400 | Duplicate TransactionId | Do not reuse TransactionId. An existing billing entry with the same transactionId will return error. |
| 1500 | Invalid Ltss Client Id | ClientId is blank in request<br>ClientId does not exist in LTSS |
| 1600 | Invalid Provider MA# | Provider MA# is blank in request<br>Provider MA# is not associated with any location |
| 1700 | Invalid Service Identifier | Service Identifier is blank in request<br>Service Identifier is invalid |
| 1800 | Exceeds allowed cap limits | Cost specified is greater than max rate defined for the service group. |
| 1900 | Exceeds allowed cap limits | Units specified are greater than max units defined for the service group. |

## HTTP Status 500

In the case of an HTTP 500 error, ensure that the header has a Content-Type set to ‘application/x-www-form-urlencoded’.